

Best Available Copy

# Encinaモニターを使ってDCEを拡張し、 クリティカルな業務システムを構築する

トランザクション処理  
分散処理  
TPモニター

- ▶ OSF/DCEは、分散コンピューティング・システムを開発・管理するための総合的なツールである。しかし、トランザクション機能を備えていないDCEだけでは、実際はクリティカルな業務システムの構築は難しい。
- ▶ DCEをベースとしたTPモニターのEncinaを組み合わせることにより、クリティカルな業務システムを構築しやすくなる。
- ▶ DCEの基本機能を説明した後、トランザクションなどの分散処理環境における障害管理機能を中心に、Encinaモニターの提供する付加機能を見ていく。

Peter J. Houston

米Transarc社

1990年、OSF (Open Software Foundation) は、大規模クライアント/サーバー・システムの開発と管理を目的とした総合的なツールの開発に着手した。その結果誕生したDCE (Distributed Computing Environment: 分散コンピューティング環境) は、通信、セキュリティ、ネーミング、タイミングといった基本的な分散コンピューティングのサービスを提供する。

DCE 1.0 は、米IBM社、米Hewlett-Packard社および米Transarc社から92年初めにリリースされた。他の大手ベンダーもその動きに従い、エンドユーザー企業は、DCEを使用する作業に着手した。共通するビジョンは、DCEを使用して、大規模クライアント/サーバー・システムを構築することである。

しかし初期のユーザーは、DCEを使っても、大規模コンピューティングにはまだ課題が残っていることに気付いた。例えば、基幹業務システム

で必要になるトランザクションなどのメカニズムはDCEの範囲を越えるもので、業務アプリケーションの構築、実行、管理に関連する多くの領域は、DCEサービスの直接的な対象ではない。

DCEについて考えるとき、DCEは終着点ではないことに注意する必要がある。終着点に向けた最初の重要なステップと見なすべきもののだ。こうした観点から、DCEは次のような一つの製品であると捉えることができる。

▷ 基礎的な分散コンピューティング・サービスを提供する。

▷ 広範囲でベンダーの制約が交わされており、一般に利用することができ、かつWindowsからOS/2、UNIX、MVSに至るプラットフォームを通じて、APIの互換性がある。

▷ 商用の分散コンピューティング・システムの要求に答えるように設計された。

こうした利点は、見過ごすことのできない重要

本記事は、Extending DCE for Business-Critical Computing with the Encina Monitor を要約したものである。©1995 Transarc Corp.

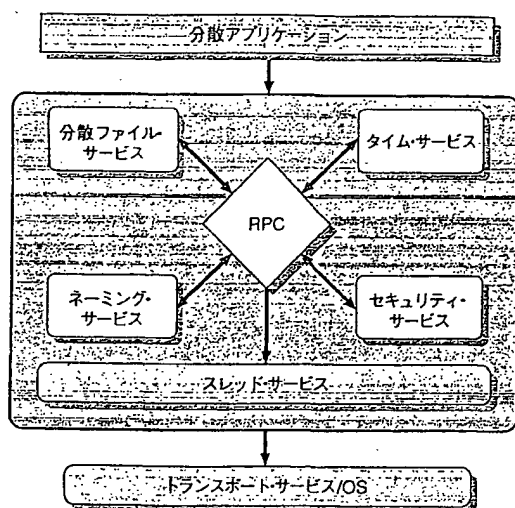


図1●DCEの構造

性をもっている。実際には、大規模クライアント/サーバー・コンピューティングの万能薬としてDCEを見るのではなく、以下のように認識すべきであろう。

▷DCE は現実的なもので、すぐ利用できる。

▷DCE は、スタンダード・ベースの分散コンピューティング・インフラストラクチャである。

▷DCE 上に構築されたサード・パーティ製品を使うとき、DCE の価値は最高になる。

最後のポイントを示す好例は、Transarc 社の Encina モニターである。

## DCEを理解する

DCE は、分散システムで必要になる機能ブロックの中のかなめとなるように設計されている。必要なサービスを思い浮かべるには、多くの部分に分けられるメインフレームを考えてみるといい。各部分間の通信のために、通信サービスが必要である。通信を保護し、アクセスをオーソライズするためにセキュリティ・サービスが必要である。

ネーミング・サービスによって、アプリケーションは、どこに必要なアプリケーションが実行されているかを見つけることができる。タイミンサービスにより、イベントの同期をとることができる。分散ファイル・サービスにより、各部分間のデータ共有が可能になる。DCE は、こうしたサービス、そしてその他の多くのサービスを提供する。

DCE の能力を評価するためには、図 1 に示した各コンポーネントの技術を理解することが重要である。その前に、まず DCE のセル・メカニズムについて説明するのがいいだろう。

## セル

DCE 環境では、個々のマシンはセルと呼ばれるクラスタにグループ分けされる。セルは、2つの基本コンセプトで設計されている。セル内のマシンは、グループとして容易に管理することができ、セル内のアプリケーションは、そのセルの範囲に必要なサービスの多くを見つけることができる。例えば、セキュリティおよびリソース・ネーミングなどの管理操作はすべて、セル・レベルで実行される。ある国に2つの異なる拠点を持つ企業は、通常、各拠点到セルをセットアップすることになる。1つのセル内のアプリケーションは、グローバル・ディレクトリ・サービスを使って他のセル内のサービスにアクセスすることができるので、DCE は管理しやすさを保ちながら、スケールを拡張する可能性を提供する。

## スレッド

DCE ベースのシステムは、複数のオペレーションを同時に実行できなければならない。ユーザー・アプリケーション・レベルでも、DCE 自体でも、そうである。例えば DCE は、多くの異なる通信ポートの使用可能性をモニターする必要がある、またアプリケーションはいくつかのオペレーションの並行実行を要求する場合がある。

多くのプラットフォームで、並行実行する基本

単位はプロセスである。しかしプロセスは一般に、多数の並行処理をサポートするためには資源の面で高価すぎる。したがってDCEは、複数のタスクを1つのプロセスで実行することが可能なスレッドを提供する。スレッドは、ユニプロセッサ・システムですぐれたパフォーマンスを提供し、SMPアーキテクチャのマルチプロセッサ・システムではより良好なパフォーマンスを提供する。

### リモート・プロシージャ・コール

どのような分散システムでも、クライアントがサーバーと通信する方法が必要である。DCE環境においては、クライアントとサーバー間の通信は、RPC (リモート・プロシージャ・コール) で実行される。

RPCは、構造化プログラミング言語がサポートしているローカル・プロシージャ・コールと似ている。異なるアドレス・スペース——典型的な例はクライアントと異なるマシン——で、コールされたプロシージャを実行するという点が違うだけである。異なるマシン間で要求を出し、応答を受ける処理は、スタブ (stub) と呼ばれるコンポーネントが実行してくれる。スタブは、インタフェース定義言語 (IDL) コンパイラと呼ばれるDCEユーティリティが生成する。DCE RPCの最大の利点は、CやPascal、COBOLでプロシージャ・コールを行う方法を知っているプログラマにとって、DCE RPCを使用するために新たに学ぶべきことが、ほとんどないことだ。

### セキュリティ

ブリッジ、ルーター、およびネットワーク・モニターを使うだけでは、ネットワーク環境で動作する業務システムに必要なセキュリティを確保するのは困難である。DCEはKerberosをベースとしたセキュリティ・サービスを提供する。

DCE環境では、特権レベルに従って、ユーザーとグループのリストを保持している。サービスにアクセスするプロセスは、2段階の処理を経る

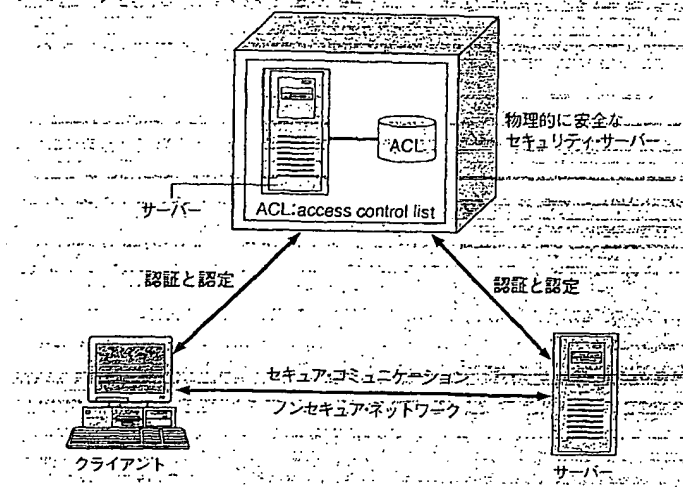


図2●セキュリティ・サービス

ことになる (図2)。

クライアントはまず、名前とパスワードを使用して、セキュリティ・サーバーにログインし、認証を受ける。認証されれば、クライアントは必要なサービスのチケットをセキュリティ・サービスに要求し、アプリケーション・サーバーにアクセスできるようになる。セキュリティ・サーバーが保持しているアクセス・コントロール・リスト (ACL) に記憶されている適正な権限をクライアントが持っていない限り、チケットは受けられない。洗練された暗号機能によって、チケットの偽装を妨ぐ。

サーバーも、セキュリティ・サーバーによって認証を受けなければならないし、またチケットのチェックが必要なので、クライアントは本当のサーバーと通信していることがわかる。サーバーも認定されたクライアントと通信していることになる。セキュリティ・サーバーを物理的に安全なデータ・センター内に置けば、セル内のマシンのセキュリティが保証される。

DCEは、パケットの暗号化やデータ・チェック・サムなどのセキュリティ機能を備えている。また、前述したすべてのセキュリティ・サービスは、ファイルや名前情報といった資源を保護する

ために DCE でも使用されている。

### セル・ディレクトリ・サービス

大規模分散システムでは、アプリケーション・サービスやファイルなどの資源を、多くの物理的に異なった場所に置くことができる。そうした資源は複数のマシンで複製することが可能である。場所は変更でき、きわめて頻繁に変更することも可能である。例えば、enter\_order アプリケーション・サーバーを、今日はマシン 1 で、明日はマシン 2 とマシン 3 で稼働させる、といったことができる。ただし、クライアント・プログラムに物理的な場所の情報を埋めこんだアプリケーションは、ただちに管理不能になる。

この問題に対処して、DCE はセル・ディレクトリ・サービス (CDS) 機能を提供する。これにより、クライアントは、場所ではなく名前で、資源の場所を参照できるようになる。CDS を正確にいうと、セル内で動作し、階層的なツリー型データベースで名前情報を維持するリポジトリである。アプリケーション・サーバーは、初期化の際に、名前と場所を CDS に入力する。クライアントがアプリケーション・サーバーを呼び出すとすると、サービスの名前が CDS に渡され、CDS は物理的な場所を返す。サービスが 2 台以上のマシンで実行される場合、クライアントは、単純なラウンドロビン方式で要求を出すことになる。名前情報へのアクセスは ACL によって保護されているので、サービスを受けることができるのは、認可されたクライアントだけである。

### 分散タイム・サービス

DCE ベースのシステムには、多くの異なるクライアントおよびサーバー・マシンが含まれているが、セル内のすべてのマシンが共通の時刻を持つことが大切である。例えば、あるマシンが 3:00 に設定され、別のマシンが 3:15 に設定されている場合、多くの問題が起こる。しかし手作業で時計を同期させるのは現実的ではない。DCE

は、この問題に分散タイム・サービス (DTS) を提供することで対処している。

DTS は指定されたマスター・マシンのグループを調べ、最もマッチする時刻を計算し、共通の時刻の値を確定する。DTS はこの値を使って収束するまで、セル内の各マシンの時計を進めるか、遅らせる。個々の時計をいっしょに進めたり遅らせたりすることはない。ロギングや同期の際に時刻を使用するアプリケーションで問題が生じるためである。収束すると、セル内の各マシンの時間のズレは 200 m 秒の範囲に保たれる。

### 分散ファイル・サービス

データの共有は、地理的に分散した環境において問題となる。DCE は、セル内およびセル間の情報を共有するために分散ファイル・システム (DFS) と呼ばれる広域ファイル・システムを提供する。DFS は既存のファイル・システムに加えて使用することができ、ユーザーから見ると区別はつかない。加えて DFS は、キャッシュによってネットワーク・トラフィックを減少させ、管理を単純化するためのシステム管理機能を持つ。

ここまでは、DCE の能力と機能を紹介してきた。もう一つ付け加えると、プログラマが DCE アプリケーションを書くために使う API がある。API は、すべてのプラットフォーム上で一定なので、DCE ベースのアプリケーションは移植性が高く、1 つのアプリケーションを多くのプラットフォーム上で使うことができる。

### DCE の範囲を認識する

DCE の機能を使うには多少の複雑さを伴うことは予想できる。DCE は分散コンピューティングのサービスを行うのに望ましい機能の集まりとして設計されている。実行可能なプログラムに対する制限は、きわめて少ない。しかし DCE サー

ビスはネットワークで利用できるものの、分散ファイル・システムを例外として、それ自体に対する作業はほとんどできない。プログラマは、クライアントとアプリケーション・サーバーを構築しなければならない。これには名前付けのシンタックスやセキュリティ、プログラミングなどの詳細を決める必要がある。例えば、サーバーに対するRPCの作業を行うには、クライアントはまず、

以下の処理を実施する必要がある。

- ▷DCE クライアントとして初期化する。
- ▷ネットワーク・プロトコルを選択する。
- ▷CDS から目的のサービスを探す。
- ▷セキュリティ・サービスの認定を要求する。

サーバー側では、プログラミングをする必要がある。例えばRPCを行うには、サーバーはまず以下の処理を実行する必要がある。

- ▷DCE サーバーとして初期化する。
- ▷CDS に名前を入力する。
- ▷セキュリティ・サービスを用いて、入ってくるRPCを有効にする。

サーバーは、自分自身のセキュリティの資格が期限切れになったときの再確認、および運転中止時におけるCDSからの名前情報の排除などの問題も扱わなければならない。DCEは、アプリケーションの実行および管理のためのアプリケーション指向の基盤構造を含むことはない。アプリケーションの起動、停止、複製、および管理などのサポート機能の構築は、プログラマに任せられる。

DCEはまた、直接的に障害管理を扱うことはない。分散環境における障害の意味はきわめて広い。故障したサーバーの再起動から通信障害に対する復旧までに渡る。例えば図3は、クライアントがDCE RPCを2つのサーバーに対して発行するシステムを示している。各々のサーバーは、異なるデータベース上で同じデータを更新することになる。これは大規模システムではきわめて一般的な構造である。

もし最初のRPCが成功し（最初のサーバーが更新を実行したことを含む）、クライアントが第

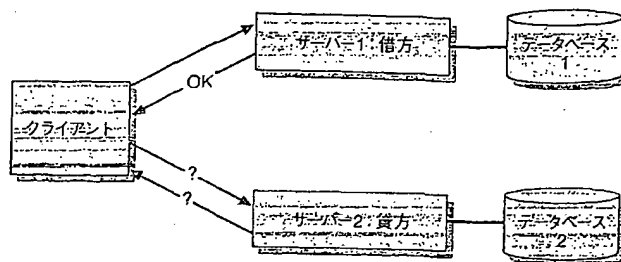


図3 複数データベースを利用するアプリケーション

2のRPCに対するタイムアウト・リターン・コードを受け取った場合、システムは未知の状態になる。クライアントから見て、サーバーがRPCを受信して応答に失敗したか、RPCが受信されなかったかを知ることはできない。前者のケースではデータは更新されているし、後者のケースでは更新されていないことになる。

こうした問題に例外処理のロジックおよび補償エントリなどの技法によって対応することは可能だが、こうしたアプローチは、すぐに操作不能になる可能性がある。必要なのは、強力な、しかし一般的な形態で問題点を扱うトランザクションなどのメカニズムである。

クリティカルな業務システムでは障害管理や、DCEの側では意図していなかったアプリケーションの実行や管理のための強力な基盤構造などの付加的な要求がある。結果として、Transarc社などのサード・パーティが付加製品を開発し、特定の領域でDCEを拡張することになる。

### クリティカルな処理のための拡張

クリティカルな業務処理には、分散システム環境に対して特別の要求をすることになる。信頼性を向上させるため、アプリケーションの複雑さを低減しなければならない。強力で弾力性に富む実行環境を提供し、アプリケーションがパフォーマンス

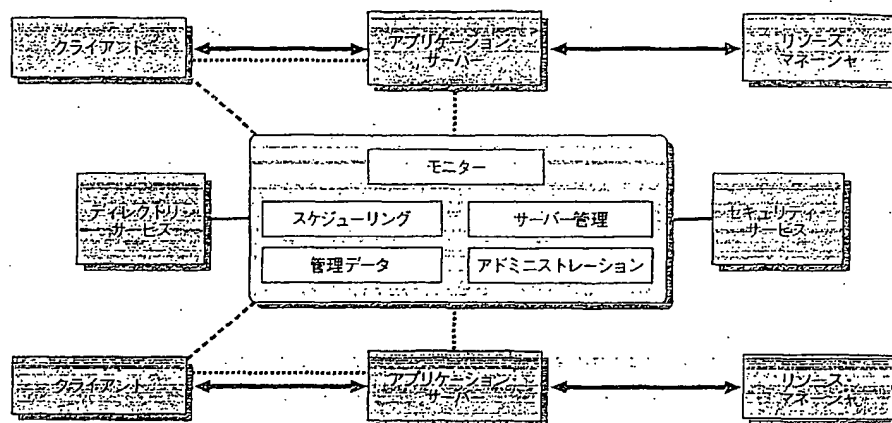


図4●Encinaモニターを用いたアプリケーション

ンスや信頼性、適用性について危険を犯すことなく、複数のマシン環境を透過的に利用できなければならない。システム管理ツールにより、オペレータは一つのまとまりとしてマシンのクラスタを見ることができなくてはならず、かつ個々のマシンの集合性も失われてはならない。Transarc社のEncinaモニターは、以下の点でDCE環境におけるこれらの要求事項にかなう。

▷DCEの上部にアプリケーション指向の基盤を提供することでアプリケーションの実行環境が得られ、Encinaモニター上のアプリケーションをシングル・イメージとして管理できる。

▷分散トランザクションを含む強力な障害管理メカニズムを、DCEに組み入れる。

▷RDB、メインフレームなどの資源に対してトランザクションの接続性を拡張する。

▷DCEベースの分散システムにおいて、並行処理と並列処理を利用できるようにする。

バー・システムをサポートしている(図4)。モニター・アプリケーションは、クライアント、アプリケーション・サーバー、リソース・マネージャにまたがる。クライアントは、プレゼンテーションとデータ確認ロジックを含み、サービスのためにアプリケーション・サーバーを呼び出す。アプリケーション・サーバーは、ビジネス・ロジックを実行し、資源マネージャをコールしてデータ関連の要求を満たす。このアーキテクチャは

▷複雑さを隠す。アプリケーション・サーバーは、インプリメンテーションの詳細をクライアントから隠す。クライアントのロジックはシンプルになりプレゼンテーションに集中できる。

▷再利用をしやすいとする。アプリケーション・サーバーとのインタフェースは十分に定義されているので、広範囲のクライアントが既存のサービスを再利用するのは簡単である。アプリケーションを簡単かつ迅速に構築でき、保守のコストも下げられる。

▷発展が容易である。アプリケーション・サーバーの配置は、外部プログラム・インタフェースが同じである限り、クライアントに影響を与えないかたちで変更できる。

▷異種のコンピューティング環境を統合する。

### Encinaモニターの アプリケーション稼働環境

Encinaモニターは、3層のクライアント/サー

Encina モニターは、人気のある RDBMS 製品を含む広範囲のリソース・マネージャに対し、2 フェーズ・コミットを含む統合を可能にする。

▷選択が可能である。標準化されたプログラミング・モデルにより、開発者は高級言語、開発ツールおよびデータ・モデルを選択できる。

▷RDBMS のパフォーマンスを改善する。

Encina モニターは、RDBMS システムからのクライアント管理とタスク・スケジュール処理のオーバーヘッドを除く。結果として、RDBMS だけの構成よりトランザクションの処理量は増す。

▷可用性とパフォーマンスを改善する。Encina モニターは、データではなくアプリケーションに注目することにより、アプリケーション・サーバーを複製することができる。複製された複数マシン上のサーバーは、可用性とパフォーマンスを改善する。

### 単純化された開発

DCE のパワーの一つは、広範囲にわたるプログラミング API のセットである。ほとんどのケースでは、開発者はビジネス・ロジックに注目し、複雑なプログラミング、実行、および管理基盤に費やす時間を最小限にしようとする。Encina モニターは、次のようなプログラム環境により、これを可能にしている。

▷プログラミングを簡略化する。Encina モニターは、プログラマのために、ネットワーク・プロトコルの選択、サーバーのルックアップ、およびセキュリティの認可など、通常は DCE クライアントの開発に関連する多くのタスクを扱う。また、DCE の初期化、セキュリティ・チェック、および稼働中のサーバーの再認可など、多くのサーバー側の操作を自動的に実行する。さらに、自動ロード・バランス調整などの機能を使うために必要なプログラミングは、きわめてわずかである。

▷Encina モニターの実行と管理の環境を統合する。Encina モニターの単純化された API を使えば、アプリケーション・サーバー・プログラ

ムは、モニター環境にプラグインできる。起動、停止、および複製などの操作は、Encina モニターによって提供されるコンソール環境を通じて実行される。パフォーマンス情報の収集やシステム・コンフィグレーションの動的な変更などの管理機能も、モニターが提供する。CDS やセキュリティ・サービスにおけるエントリの管理といった DCE タスクは、自動的に実行される。

▷特定の機能への依存を少なくする。Encina モニターは、プログラマが標準の高級言語、API、およびプロトコルを使用してアプリケーションを構築するのを助け、ストアド・プロシージャといった特定の機能を使わなくてもよくする。

▷標準開発ツールの使用を容易にする。プログラマは標準的な開発、デバッグおよびソース・コード管理ツールを使用することができ、これにより選択の幅が広がり、依存性を低減する。

こうした機能はすべて、DCE フレームワークの中で提供される。

### アプリケーションの実行

アプリケーションを開発すると、それが稼働することのできる十分な機能を備えた実行環境をもたせることが大切になる。実行環境は、アプリケーション・サーバーの起動および停止のための基本インタフェースから、動的アプリケーション・サーバーの複製といった、より複雑な機能までに渡る。加えて実行環境は、クリティカルな業務システムをサポートできるように弾力性に富んだものでなくてはならない。DCE 自体は実行環境を提供しない。しかし、Encina モニターは DCE ベースの環境に以下の機能を付加する。

▷サーバーの起動、停止、および静止などのタスクを行う基本的なアプリケーション・サーバー管理ユーティリティ。

▷システムがオンライン中でもアプリケーション・サーバーを複数マシン上に複製する機能。

▷利用可能なマシンに対する自動ロード・バランス機能。

▷システム管理者が作成する DCE アクセス・コントロール定義の自動発効。

▷2 フェーズ・コミットを透過的に実行することにより、分散更新時に異なるリソースにまたがって確実にデータの整合性を保つ機能。

▷すべてのサブシステムとアプリケーション・サーバーが稼働していることを確認するための、定期的なシステム・ヘルス・チェック。故障の場合、Encina モニターは自動的に故障したコンポーネントの再起動を試みる。

▷故障したリソースが復帰してから、データベース・トランザクションの同期化を含むシステム状態の自動復旧。

Encina モニターを利用したアプリケーションを構築すれば、これらの機能を自動的に利用できる。アプリケーション稼働環境そのものは、CDS やセキュリティ・サービスなどの DCE コンポーネントを使って構築されており、DCE が提供する機能を利用することができる。例えば、複製およびロード・バランス情報は、CDS ネーム・スペースに保存され、すべてのセキュリティ操作は DCE セキュリティ・サービス上で行われる。

## アプリケーション管理

DCE は、大規模クライアント/サーバー・コンピューティングのためのプラットフォームを提供するように設計された。すべての DCE システムは、何千人ものユーザーを扱い、地理的に異なる多くの場所に範囲を拡張するように設計されている。しかし、大規模クライアント/サーバー・システムは扱いが最も難しいものとされており、DCE 上に構築されるシステムも例外ではない。DCE は、CDS やセキュリティ・サービスなどの、それ自身のサブシステムを管理するユーティリティを備えているが、DCE ベースのアプリケーションのシステム管理の要求に答えるものではない。必要なのは、DCE ベースの大規模クライアント/サーバー・アプリケーションを管理することに焦点を合わせた環境である。Encina モニ

ターは、下記のものを提供する。

▷分散 Encina モニター・アプリケーションのシングル・イメージ・システム管理。管理者は、ネットワークのどの場所においても稼働しているすべての Encina モニター・アプリケーションとシステム・コンポーネントのコンフィグレーションおよび状態を迅速に決定することができる。また管理者はネットワーク上のマシンから、セキュリティ管理、アプリケーションの起動と停止、およびオンライン・コンフィグレーションの変更を行うことができる。

▷関連するマシンから自動的に収集したシステム・ステータスとパフォーマンス情報を確認する機能。

▷オンライン・システム管理機能。サーバーのコンフィグレーション、アプリケーション・サーバーの追加コピー、バックアップとログ・ファイルのリストアといった管理作業は、サービスを中断することなしに行える。

▷クライアントの特権レベルを、中央で管理する機能。これは Encina モニターの一般的なセキュリティ・モデルである。

▷あるバージョンから他のバージョンへのクライアントの段階的な移行を可能にするオンライン・バージョン・アップ機能。アプリケーション・サーバーの新しいバージョンは、古いアプリケーションが作動している間に起動させられる。

▷Encina の管理機能を他の管理環境に統合することのできるすべての API。

## Encina モニターの 障害管理メカニズム

集中型システムの開発者は、何年も前に、システム障害の取扱い方を学んだが、分散システムの出現により、従来技術の対象ではないまったく新しいタイプの障害が発生することになった。特に大規模分散システムを対象とするものは免疫がな



く、実際に故障を起こすきっかけを自ら作り出す。

これには無視できない2つの領域がある。

▷DCEには、トランザクションの作業単位というコンセプトがない。トランザクションなしでは、通信故障やデータベースの更新の失敗などのイベントは、DCEが直接扱うことができない分散環境に置かれることになる。

▷アプリケーション・サーバー、データベース、およびモニター自体といった実行基盤内の障害の影響。

Encina モニターは、DCE RPC メカニズムに対してトランザクション・サポートを加えることにより、これらの双方の懸念に対して対処する。また実行基盤内で、障害検知と修正サポート機能を提供する。これらの機能について、下により詳しく見ていこう。

### 分散トランザクションのサポート

トランザクションの作業単位というコンセプトは、多くのクリティカルな業務システムに存在する。トランザクションの定義は複雑だが、実際上トランザクションはアプリケーションに対して2つの有意義な利点を提供する。1つのトランザクション内で実行される作業は次の通りである。

▷一つの処理単位として完了するか、またはトランザクション内における処理を実行しないようにロールバックされる。多くのアプリケーションでこの特性が必要である。例えば、在庫の移動と出荷は1つの単位として実行する。

▷システム内の他の活動から分離されている。

これは、2つのアプリケーションが同時に同じデータにアクセスしようとする場合、トランザクションが必ず1回に1つのアプリケーションだけにアクセスさせるということを意味する。2番目のアプリケーションは、最初のアプリケーションが完全に終了するまで待たなければならない。この保証がないと、開発者は複雑な保護ロジックをプログラムに書き込み、同時アクセスの影響を検知し、それを修正しなければならない。

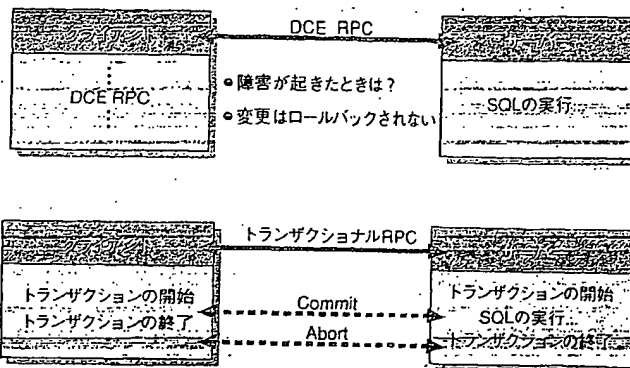


図5 ⑤ トランザクショナルRPC

DCEはトランザクションのコンセプトを提供しない。そのため、DCEベースのシステムでは2通りの方法で影響が出る。

▷前述したように、DCE RPCについての通信不良により、データが未知またはあいまいな状態になる可能性がある。

▷多重更新の際に、1つ以上のデータ資源が、そのデータを適正に更新しない場合、他のデータ資源で完了している処理にアンドゥ操作を行うことがきわめて難しくなる可能性がある。

Encina モニターは、DCEに対してトランザクション・サポート機能を追加することにより、こうした問題を解決する。DCE RPCは、トランザクション属性を付加し拡張されたトランザクショナルRPC (TRPC) になる。

Encina モニター環境において、クライアントの開発者たちは、単純なキーワードを使用してトランザクションとして実行されるビジネス・ロジックの始めと終わりを定義した。そして、TRPCを用いてアプリケーション・サーバーを呼び出した(図5)。これにより、サーバー上で実行されるすべてのデータ更新は、クライアントのトランザクション・ステートメント内からアクセスされるグローバル・トランザクションの一部として実行されることになる。すべての更新が成功すると、

それはユニットとして完了する。あるアプリケーション・サーバー内でアボートが発生するか、または通信が失われると、Encina モニターはすべての変更を自動的にロールバックする。

トランザクション・サポートは追加されたが、クライアントとアプリケーション・サーバーとの間のすべてのコールは、やはり通常の DCE RPC で送られる。プログラマから隠されるのは、トランザクション・コンセプトの保持、バッファ管理やデータ転換といったシステム内通信の取扱い、および 2 フェーズ・コミット・プロトコルの調整である。TRPC を作成するには、開発者はトランザクション IDL ファイル (DCE IDL ファイルにきわめて似ている) を作成し、Encina モニターの TIDL ジェネレータを実行し、結果としてのコード・スタブをアプリケーション・プログラムにリンクさせればよい。

#### 故障の検知と修正

DCE ベースのシステム内で障害が発生するきっかけは、アプリケーション実行環境それ自体であることもある。例えば、アプリケーション・サーバーがクラッシュすると、システム内のすべての作業は停止し、サーバーの再起動待ちとなる。アプリケーション実行環境は、故障を検知し、かつ修正を実行する必要がある。自動化できていない場合、オペレータが変更しなければならない。アプリケーションの実行のために DCE が提供する機能はきわめてわずかなので、これがしばしば問題を発生させる。Encina モニターは 2 通りの方式で DCE ベースの故障検知と修正に対処する。

▷自動化されたモニターと再起動: Encina モニターは、Encina モニター・セル内の各マシンで実行されるアプリケーション・サーバーとモニターのプロセスに絶えず注意している。モニターは、不良のプロセスを再起動させようとし、不成功の場合には、オペレータ・コンソールに知らせる。またアプリケーション・サーバー・プロセスが再起動すると、マシンにおいて進行中であった

トランザクションは自動的に回復し、再び同期がとられる。

▷自動化されたクライアントの再ルーチング: Encina モニターは、自動的にクライアントのリクエストを、利用可能なアプリケーション・サーバーに対してルーティングする。クライアント・プログラムは単純に TRPC コールを行うだけで、すべてのルーティング・ロジックは内部的に操作される。サーバーが故障した場合、モニターはそれが再び使用可能になるまで、そのサーバーに対するルーチングの要求を停止する。

#### Encina モニターが提供するアプリケーション資源

Encina モニターが提供するトランザクションの利点の 1 つは、単純かつ構造化された形式でさまざまな故障を検知し扱う機能である。しかしながら、モニターのトランザクションが有意義なのは、X/Open によって指定される XA インタフェースなどの、トランザクショナル・インタフェースをサポートするリソースと共に使用するときだけである。有名な RDBMS などは XA サポートを提供し、したがって Encina モニター環境に容易に統合できる。一方、大規模クライアント/サーバー・システム内のトランザクショナルな統合の必要性は、RDBMS 製品をはるかに超えたもので、レコード形式のファイル・システムおよびメインフレーム・ゲートウェイなどの、付加的なタイプのリソースも要求する。こうした要求に答えるため、Encina ファミリーにはいくつかのトランザクショナルなリソースを含んでいる。

▷回復可能キューイング・サービス (RQS): 十分な機能を備えたデータ・キューイング・サービス

▷ピア・ツー・ピア通信 (PPC) システム: LU 6.2 ベースのメインフレーム・ゲートウェイ

▷構造化ファイル・サーバー (SFS): 強力な

レコード指向ファイル・システム。

これらの資源のすべては、Encina モニター環境に統合されている。各々についての詳細を下に示す。

### 回復可能キューイング・サービス

「オール・オア・ナッシング」の特性を維持するために、ほとんどのビジネス・トランザクションは、トランザクションを行う前に、すべてのステップが完了する構造になっている。Encina モニター環境においては、通常トランザクションはクライアントで始まり、ビジネス・ロジックのピースを実行するアプリケーション・サーバーに対する多くの TRPC を含んでいる。TRPC が行われると、クライアントは結果が戻るのを待ち、それから次のステップに進むことになる。このスタイルは同期通信として知られている。

しかし、分散システムにおいて発生する障害や遅延のため、アプリケーション・サーバーに対する同期コールを基礎にした複雑なトランザクションを構築することは必ずしも実際的ではない。例えばアプリケーション・サーバーへの TRPC が復帰までに長時間を要する場合、トランザクショナル・グリッドロックの形態が発生する可能性がある。これは、関連アプリケーション・サーバーすべてにおいて、トランザクションによってアクセスされるリソース全体が、すべての処理要素が終了するまで保持されるために発生する。保持されたリソースにアクセスする必要がある他のトランザクションは、逆に自己のリソースを保持することになる。

この問題に対処する1つの強力な方法は、キューを使用することである。キュー・ベースのモデルでは、クライアントは通常データをエンキュー（待ち行列を作成）し、これが後にデキュー（待ち行列を削除）され、アプリケーション・サーバーによって処理される。データがキュー内にある間、クライアントとアプリケーション・サーバーとの間に同期結合はない。Encina の RQS は、

次の場合にトランザクションを使用してキュー処理を結合することにより、キューの価値をさらに高める。

▷RQS のエンキュー・オペレーションがトランザクショナルな作業のユニットの一部として実行される場合、すべてのエンキューがユニットとして完了するかロールバックしなければならないトランザクションの一部として扱われるということの意味する。例えば、トランザクションが RDBMS テーブルとエンキュー・オペレーションの更新を含み、かつ RDBMS の更新がなされない場合、エンキューはアボート処理の一部としてロールバックされることになる。

▷エンキュー処理トランザクションが行われた後は、RQS に記憶されているエレメントは失なわれたり、壊れたりすることはない。これは、ディスク・ミラー処理およびログ・ベースの回復メカニズムなどの技術を使用することにより実現される。

▷RQS デキュー・オペレーションもトランザクショナルである。例えば、トランザクションがエレメントのデキュー処理を行い、かつ処理中にアボート条件に出会うと、エレメントは自動的にキューに戻る。

トランザクショナルな性質のために、RQS キューを使用して、完了しなくてはならない複雑な長時間稼働するオペレーションを、独立して処理することのできる小さな多くのピースに区分することが可能である。

また、エレメントがエンキュー処理されることがとデキュー処理されることが間に同期的な依存性はないので、RQS はキュー・エントリを処理するために、同期 TRPC を使用していても、Encina モニター・アプリケーションを非同期で動作させることができる。

回復可能なキューは、他の資源を使用して分散トランザクションをサポートする全体的な環境に統合されてこそ意味がある。Encina モニター環境は、下記の機能を提供する。

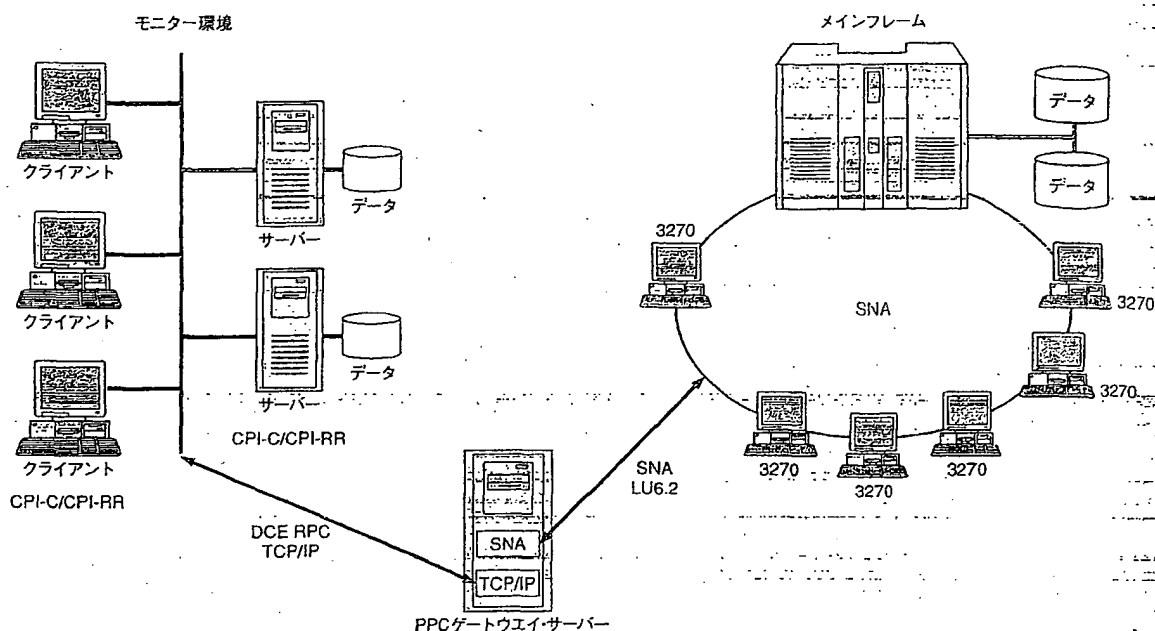


図6●メインフレームとの統合

トランザクションに含まれる他の資源を使う作業と、エンキューおよびデキューの作業を調整するトランザクション管理サービス。

アプリケーション・サービスのエンキューおよびデキュー作業の範囲内で、多くの異なる種類の資源にトランザクショナルにアクセスする能力。例えば、デキューイング・サービスの一部は、メインフレーム・ベースのサービスをコールすることができる。Encina モニターがないと、メインフレームを作業のユニットの一部として呼び出すことができず、アポート条件はメインフレーム・データの完全性について妥協の余地を残すことになる。

アプリケーションの複製および、しきい値のトリガー動作などの他の Encina モニターの機能を統合する。アプリケーションの複製により、複数のマシンでデキューを行い、エントリを処理して個々のキューがボトルネックにならないようにすることができる。しきい値のトリガーにより、RQS キューが特定のエントリのレベルに達したときに、アプリケーションを効果的に呼び出すこ

とができる。

障害時におけるキュー復旧サービス。

一般に、大規模クライアント/サーバー・システムの開発者は、信頼性の高いキューにするために、多くの付加的な使用法を見い出している。実際に、延期された処理またはオンライン・システムにおける要求のピークと谷の部分の緩衝などを確実に行うといった分散コンピューティングの最も難しい点は、キューを使用して解決されている。Encina の RQS は、信頼性の高いキューイング技術が必要なアプリケーション実行のための強力なツールを提供する。

## ピアツーピア通信システム

94 年の MVS オープン・エディション・プロダクトのリリースと共に、米 IBM 社は、メインフレームに対するネイティブ DCE の接続性を拡張した。これは、一方では分散コンピューティングにおけるマイルストーンとも言うべきものだ。クライアントは、実質的にどのハードウェア・プ

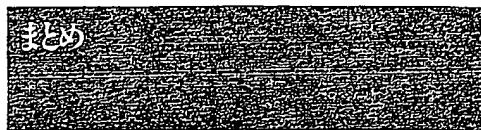
ラットフォームからでも DCE RPC を使用するメインフレーム上で稼働するアプリケーション・サーバーと対話することができる。その逆も成り立つ (図 6)。しかし、MVS DCE はトランザクション・サポートを行わない。これは、すべての使用法に対する制限ではなく、多くの DCE ベースのアプリケーションは、資源の完全性を保つ方式でメインフレームと通信することになる。これは Encina の PPC プロダクトによって実現する。

Encina の PPC プロダクトにより、Encina モニター・アプリケーションは、ピア・ツー・ピア方式で他のモニター・アプリケーションおよびメインフレーム上のアプリケーションと通信することが可能になっている。PPC エグゼクティブ・プロダクトは、TCP/IP における Encina モニター・アプリケーション間におけるピア・ツー・ピアの通信を実行するために、SAA および X/Open スタンドアードの CPI-C プログラム・インタフェースを提供する。オープンシステムとメインフレームの世界の橋渡しをするために、PPC ゲートウェイ/SNA は、TCP/IP から SNA へのゲートウェイを提供し、これにより CPI-C を使用して書かれたモニター・アプリケーションは、メインフレーム上で改造されていない LU 6.2 の実行可能なアプリケーションと透過的に対話することができる。

PPC はまた、Synclevel Syncpoint (Synclevel 2) 通信を可能にする標準 CPI-RR プログラミング・インタフェースを使用して、その CPI-C のサポートを拡張している。Synclevel 2 は、Encina の 2 フェーズ・コミット・プロトコルを、CPI-C および CPI-RR で書かれた他の Encina モニター・アプリケーションに対して透過的に実行することができる。PPC ゲートウェイ/SNA も、Synclevel 2 をサポートしており、モニター・アプリケーションとメインフレームとの間の十分なトランザクションの保証を透過的に提供する。例えば、単一の Encina モニター・トランザクションは、メインフレーム上の CICS アプリケ

ーションを呼び出すだけでなく、UNIX サーバー・マシンにおける RDBMS を更新することができる。RDBMS の更新が行われない場合、CICS によって実行される作業は、自動的にロール・バックされる。この逆も成り立つ。

LU 6.2 に対する PPC のサポートも双方向であり、Encina モニター・アプリケーションを分散システムにおいてクライアントおよびサーバーのどちらとしても作動させることができる。例えばモニター・アプリケーションはメインフレーム・ベースのアプリケーションによって提供されるサービスに対して要求を行うことができ、またメインフレームからの要求のサービスを行うことができる。LU 6.2 の性質のために、モニター・アプリケーションもメインフレーム・アプリケーションも、どこで要求がサービス処理されているのかには関知しない。



大手のユーザー企業やハードウェア・ベンダーのサポートを得て、DCE は大規模クライアント/サーバー・システムのスタンダードになっている。多くの組織が下記の目的で、DCE をベースとする付加価値製品を使用する必要があるということは明かである。

▷ネイティブ DCE プログラミングの複雑性にかかわらないようにする。

▷クリティカルな業務アプリケーションをサポートすることのできる実行および管理環境を提供する。

▷トランザクション・サポートやメインフレームの接続性など、DCE の範囲を越える大規模クライアント/サーバー・システムにより必要な機能を追加する。

Encina モニターは、分散コンピューティングの DCE 様式を保ちつつ DCE を拡張する。

【受入日】 19981204

【情報館受入日】 19950714

【C Sターム】 CZ01、CC03、CC06、CC07、CZ07、CC08、DZ01、DZ02、DZ04、DZ06、DD07、DZ07、DD09、DZ09、DD14、DZ14、EE01、EZ03、JZ02、JJ03、JZ03、JJ05、JZ05、JZ07、JZ27、KK60、LZ02、LL03、LZ03、LL05、LL09

【フリーワード】 Encinaモニター、分散コンピューティング・システム、開発・管理、TPモニター、障害管理機能、APIの互換性、スレッド、リモート・プロシージャ・コール(RPC)、クライアント、サーバー、セキュリティ、パスワード、認証、暗号機能、セル・ディレクトリ・サービス(CDS)、大規模分散システム、階層的ツリー型データベース、ラウンドロビン方式、分散タイム・サービス(DTS)、ロギング、同期、分散ファイル・サービス、分散ファイル・システム(DFS)、ネットワーク・プロトコル、信頼性の向上、並行処理、並列処理、デバッグ管理ツール、アプリケーション管理、分散トランザクション、故障の検知・修正、キューイング・サービス、ピアツーピア通信システム、学術研究機関、PPCゲートウェイ/SNA、インタフェース定義言語(IDL)コンパイラ、セル、OSF/DCE、ACL、API、RDB、RDBMS、TCP/IP、Windows、OS/2、UNIX

【許諾レベル】 2X

【著者群】

【著者名】 Peter J. Houston

【著者所属】 米Transarc

【論文タイトル】 Encinaモニターを使ってDCEを拡張し、クリティカルな業務システムを構築する

【資料タイプ】 技術雑誌(国内)

【ジャーナル】

【ジャーナルタイトル】 日経オープンシステム 第28号 NIKKEI OPEN SYSTEMS

【発行者名】 日経BP社 Nikkei Business Publications, Inc.

【開催日・発行日】 19950715

第28号

【頁】 291~303

【ISSN】 0918-581X

## 拒絶理由通知書

特許出願の番号	特願 2001-151899
起案日	平成16年10月27日
特許庁審査官	宮司 卓佳 9555 5B00
特許出願人代理人	坂口 博(外 1名) 様
適用条文	第29条柱書、第29条第2項、第36条

この出願は、次の理由によって拒絶をすべきものである。これについて意見があれば、この通知書の発送の日から3か月以内に意見書を提出して下さい。

### 理 由

A. この出願は、特許請求の範囲の記載が下記の点で、特許法第36条第6項第2号に規定する要件を満たしていない。

#### 記

(1) 本願の請求項1乃至14には、「・・・することができる少なくとも1つのプロセッサ」(例：請求項1の「エンドユーザ・トランザクションを認識することができる少なくとも1つのプロセッサ」)と、実行可能な機能によってプロセッサを特定する記載があるが、これら実行可能な機能によって特定されるプロセッサとは、どのようなものであるのかを明確に把握することができない。

(現在の請求項1乃至14の記載は、プロセッサの物理的特徴を記載したものとも、プロセッサをソフトウェアとハードウェア資源とが協働した具体的手段として記載したものともいえない点に注意されたい。)

(2) 請求項6の「前記選択された機能値は、1つ以上のエンドユーザ・トランザクションのクラス・ラベルを形成する」との記載は、「機能値」なる値が「形成する」という動作を行うように解されるが、「機能値」が「形成する」という動作を行うということはどのようなことを意味するのかが日本語として明確でない。

請求項20の、「前記選択された機能値は、1つ以上のエンドユーザ・トランザクションの代表的なクラス・ラベルを形成する」との記載についても同様である。

(3) 請求項8の「(i) 前記モデルが前記ラベルされた1つ以上のリモート・プロシージャ・コールから学習され、かつ前記1つ以上のエンドユーザ・トランザクションでラベルされている1つ以上の選択された機能を計算すること」との

記載において、「モデル」、「選択された機能」、「(「プロセッサ」)」と「計算すること」との関係が日本語として明確でないため、該記載が何を意味するのかを明確に把握することができない。

(4) 請求項1-4の「プロセッサは、さらに前記エンドユーザ・トランザクションをマーキングする際に使用するために前記クライアント・ワークステーションに対してマーキング・エージェントをダウンロードする」との記載は、「マーキング・エージェント」がサーバ又はクライアント・ワークステーションのどちらから、どちらへ移動することを意味するのかが日本語として明確でない。

請求項28に記載の「前記生成ステップは、前記エンドユーザ・トランザクションで使用するために前記クライアント・ワークステーションに対してマーキング・エージェントをダウンロードする」についても同様である。

(5) 請求項16に記載の「前記抽出エンドユーザ・トランザクション」は、何を指すのかが明確でない。

(本願において「エンドユーザ・トランザクション」と「エンドユーザ・トランザクション機能」とは異なる意味で使用されていることに十分に注意されたい。

)

(6) 請求項22の「前記モデルが前記ラベルされたリモート・プロシージャ・コールから学習され、前記1つ以上のエンドユーザ・トランザクションによってラベルされている1つ以上の選択された機能を計算するステップ」とは、どのようなステップを意味するのかが日本語としても技術的にみても明確でない。

(7) 請求項27の「エンドユーザ・トランザクションによってラベルされて前記トレーニング・データとしての役割を果たすリモート・プロシージャ・コールを生成するために前記サーバで受信した前記1つ以上のリモート・プロシージャ・コールによって前記エンドユーザ・トランザクションを相関させるステップ」との記載において、リモート・プロシージャ・コールによってエンドユーザ・トランザクションを相関させるとはどのような事項を意味するのかが技術的に明確でない。

(8) 請求項29乃至31は、「1つ以上のプログラムが含まれる機械読み取り媒体を含むことを特徴とする製品」につき特許を受けようとする発明としたものであるが、該「製品」なる語によって特定されるものの範囲を明確に把握することができないから、請求項29乃至31に係る発明の範囲は明確でない。

よって、請求項1-14、16-20、22-31に係る発明は明確でない。



B. この出願は、発明の詳細な説明の記載が下記の点で、特許法第36条第4項に規定する要件を満たしていない。

記

(1) 本願の請求項2及び16の「1つ以上のリモート・プロシージャ・コールを複数のエンドユーザ・トランザクション・セグメントにセグメント化」との記載は、リモート・プロシージャ・コール1つを2以上のエンドユーザ・トランザクション・セグメントに分けることを文言上包含するものと認められる。

しかし、リモート・プロシージャ・コール1つを2以上のエンドユーザ・トランザクション・セグメントに分けることをどのように実現すればよいのかは、本願明細書及び図面の記載全体並びに出願時の技術常識を参酌しても全く不明である。

(本願実施例においては、「RPCストリーム」をセグメント化していることに注意されたい。)

(2) 本願の請求項13の「i) 前記クライアント・ワークステーションで生成される1つ以上のエンドユーザ・トランザクションの開始及び終了をマークするために前記クライアント・ワークステーションを生成すること・・・ができる少なくとも1つのプロセッサ」とは、どのようにして実現すればよいのかが技術的に明確でない。

(現在の記載は、「プロセッサ」が「クライアント・ワークステーション」自体を生成可能であると解される。)

よって、この出願の発明の詳細な説明は、当業者が請求項2-6、13、14、16-20に係る発明を実施することができる程度に明確かつ十分に記載されていない。

C. この出願の請求項1-28に記載されたものは、下記の点で特許法第29条第1項柱書に規定する要件を満たしていないから、特許を受けることができない。

記

請求項1について;

本願の請求項1に係る「エンドユーザ・トランザクション認識装置」は、プロセッサ及びメモリを有する装置ではあるものの、「1つ以上のリモート・プロシージャ・コールに関連した1つ以上のエンドユーザ・トランザクションを認識することができる」プロセッサというだけでは、エンドユーザ・トランザクションの認識という処理を行う装置がソフトウェアによる情報処理がハードウェア資源を用いて具体的に実現されているものとみることができない(プロセッサ等がどのように動作することによって「エンドユーザ・トランザクションの認識」が実現されるのかが特許請求の範囲の記載上何ら具体的でない)から、本願の請求項

1に係る「エンドユーザ・トランザクション認識装置」は、ソフトウェアによる情報処理がハードウェア資源を用いて具体的に実現されたものとはいえず、「自然法則を利用した技術的思想の創作」に該当するものとはいえない。

請求項2乃至7について；

本願の請求項2乃至7に係る「エンドユーザ・トランザクション認識装置」についても、「セグメント化」、「エンドユーザ・トランザクション機能を抽出」、「モデルを用いて各セグメントに対するエンドユーザ・トランザクション機能から1つ以上のエンドユーザ・トランザクションを計算する」（以上請求項2）、「対応する抽出された機能に対して、ベイズ・ネット・モデルに関連する機能値を割り当てこと」、「各機能値に対して後の確率分布を見出すこと」、「後の確率分布に基づいて1つ以上のエンドユーザ・トランザクションを表す機能を選択する」（以上請求項4）、「結果の少なくとも一部が品質メトリックと比較される」（請求項7）といった、エンドユーザ・トランザクション認識動作は、プロセッサ等のハードウェア資源がどのように動作することを意味するのが特許請求の範囲からは明らかでないから、本願の請求項2乃至7に係る「エンドユーザ・トランザクション認識装置」についても、ソフトウェアによる情報処理がハードウェア資源を用いて具体的に実現されたものとはいえず、「自然法則を利用した技術的思想の創作」に該当するものとはいえない。

請求項8乃至12について；

本願の請求項8乃至12に係る「モデル生成装置」は、モデルを生成するという処理がプロセッサ等のハードウェア資源とソフトウェアとがどのように協働することにより実現されているのかが特許請求の範囲は何ら具体的に記載されていない（プロセッサ等のハードウェア資源が、ラベルされたR P C等の情報に基づいてどのように学習・モデル構築を行うのかについて、該請求項の記載は何ら具体的にない）から、本願の請求項8乃至12に係る「モデル生成装置」についても、ソフトウェアによる情報処理がハードウェア資源を用いて具体的に実現されたものとはいえず、「自然法則を利用した技術的思想の創作」に該当するものとはいえない。

請求項13及び14について；

本願の請求項13及び14に係る「トレーニング・データ自動生成装置」において、「クライアント・ワークステーションを生成すること」及び「エンドユーザ・トランザクション・マークを前記サーバで受信された前記1つ以上のリモート・プロシージャ・コールと相関させること」が「できる」という記載では、これらの処理がプロセッサによって具体的に実現されているものとみることができないから、本願の請求項13及び14に係る「トレーニング・データ自動生成装置」についても、ソフトウェアによる情報処理がハードウェア資源を用いて具体

的に実現されたものとはいえず、「自然法則を利用した技術的思想の創作」に該当するものとはいえない。

請求項15乃至28について;

本願の請求項15乃至28に係る方法は、全体としてみればエンドユーザ・トランザクション認識、モデル生成またはトレーニング・データ生成のための人為的取り決めに基づくものであって、しかも、該方法全体がプロセッサ等のハードウェア資源を用いて実現されている等の自然法則を利用したといえる点も見出せないから、これら方法は全体として自然法則を利用したものとはいえず、「自然法則を利用した技術的思想の創作」に該当するものとはいえない。

総括;

以上、請求項1-28に係る発明は、「自然法則を利用した技術的思想の創作」に該当するものではないから、特許法上の「発明」に該当するものとはいえず、特許法第29条第1項柱書に規定する要件を満たすものではない。

D. この出願の請求項1、13-15、27、28に係る発明は、その出願前日本国内又は外国において頒布された下記の刊行物に記載された発明又は電気通信回線を通じて公衆に利用可能となった発明に基いて、その出願前にその発明の属する技術の分野における通常の知識を有する者が容易に発明をすることができたものであるから、特許法第29条第2項の規定により特許を受けることができない。

#### 記

先行技術文献1: Peter J. Houston, 「Encina モニターを使ってDCEを拡張し、クリティカルな業務システムを構築する」,  
日経オープンシステム, 日経BP社,  
1995年7月15日, 第28号, p. 291-p. 303

#### 備考

先行技術文献1の第299頁及び同300頁に記載された「トランザクショナルRPC」(TRPC)のように、クライアント-サーバ間でトランザクション属性(本願における「エンドユーザ・トランザクション・マーク」に対応)をRPCに付加することによって、クライアント-サーバ双方で処理単位であるトランザクション(本願における「エンドユーザ・トランザクション」に対応)を認識可能とする技術は周知であり、本願の請求項1及び15に係る発明は該周知の技術から当業者が容易に発明をすることができたものである。

また、トランザクション属性が付加されたRPCを単にトレーニングデータとして使用しようとする、及び、RPCへのトランザクション属性の付加をエージェント等のプログラムに基づき実行するよう構成することはいずれも当業者にとり格別の技術的困難性を要することではないから、本願の請求項13、14

、27及び28に係る発明についても該周知の技術に基づき当業者が容易に想到し得たものである。

拒絶の理由が新たに発見された場合には拒絶の理由が通知される。

---

先行技術文献調査結果の記録

・調査した分野   IPC第7版   G06F15/00, 12/00  
  G06F13/00

DB名

・先行技術文献

この先行技術文献調査結果の記録は、拒絶理由を構成するものではない。

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☒ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINE(S) OR MARK(S) ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**